

Апгрейд и рефакторинг
PHP-проектов
— теперь это просто
Александр Володин
php
PHP Russia
2022

Обсудим, как упростить и ускорить

- апгрейд кода проекта
- обновление пакетов
- архитектурный рефакторинг

О чём будет доклад?



Rector

Пых

Пыхарь, используешь ли ты Rector?

Анонимный опрос

43% Что?

39% Слышал про такое, не пробовал.

5% Использую только в pet-проектах.

3% Пробовали на работе, не удалось внедрить. В комментариях расскажу, почему.

10% Успешно используем в рабочем проекте.

1 503 голоса

О себе



Александр Володин

- Backend-разработчик
- 8+ лет работаю с PHP и Symfony
- Когда-то был тимлидом, а потом подустал..
- Работаю в Skyeng в отделе маркетинга: разрабатываю разные крутые штуки для привлечения клиентов

Часть первая: делаем рефакторинг

Быстро. Надежно. Недорого.



Немного драмы

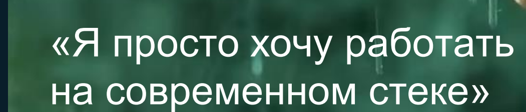
Здравый смысл:

«Зачем вы страдаете, мистер Андерсон?»

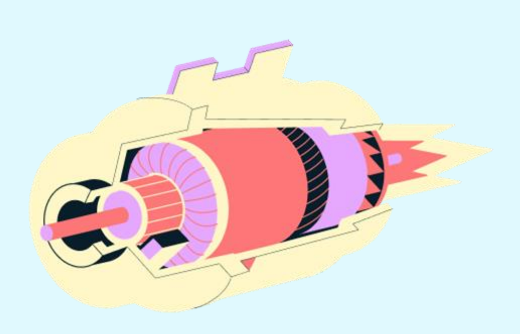
«Во имя чего?»

«Зачем рефакторите то, что работает?»

Это я в каком-то легаси-проекте ...




Очередной
легаси-проект
Ah shit, here we go again.
PHP Russia
2022



Мои классные проекты

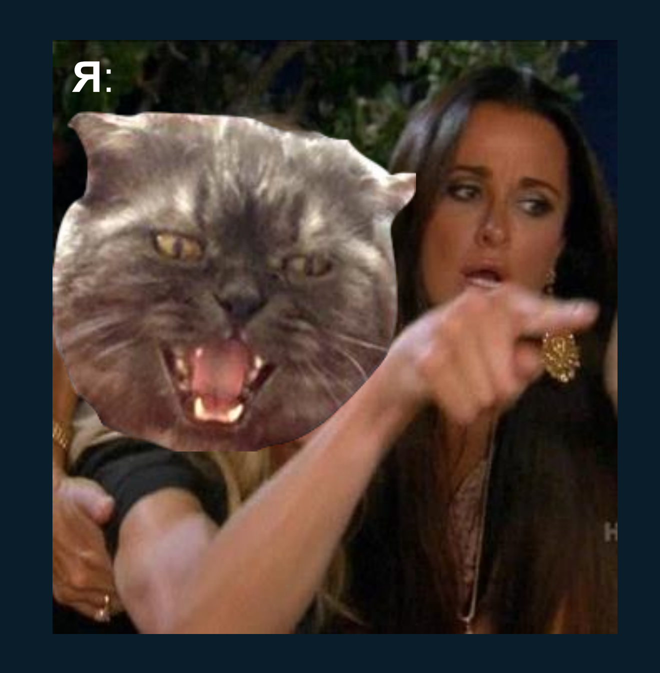
- идеальная архитектура
- современный стек: PHP 8, Symfony 5
- код-стайл - просто загляденье



Чужие легаси-проекты

- кто так сервисы называет?!
- ээх.. где ты, типизация свойств, я уже скучаю
- и функций ещё таких нет…
- хоть код-стайл-то можно было добавить





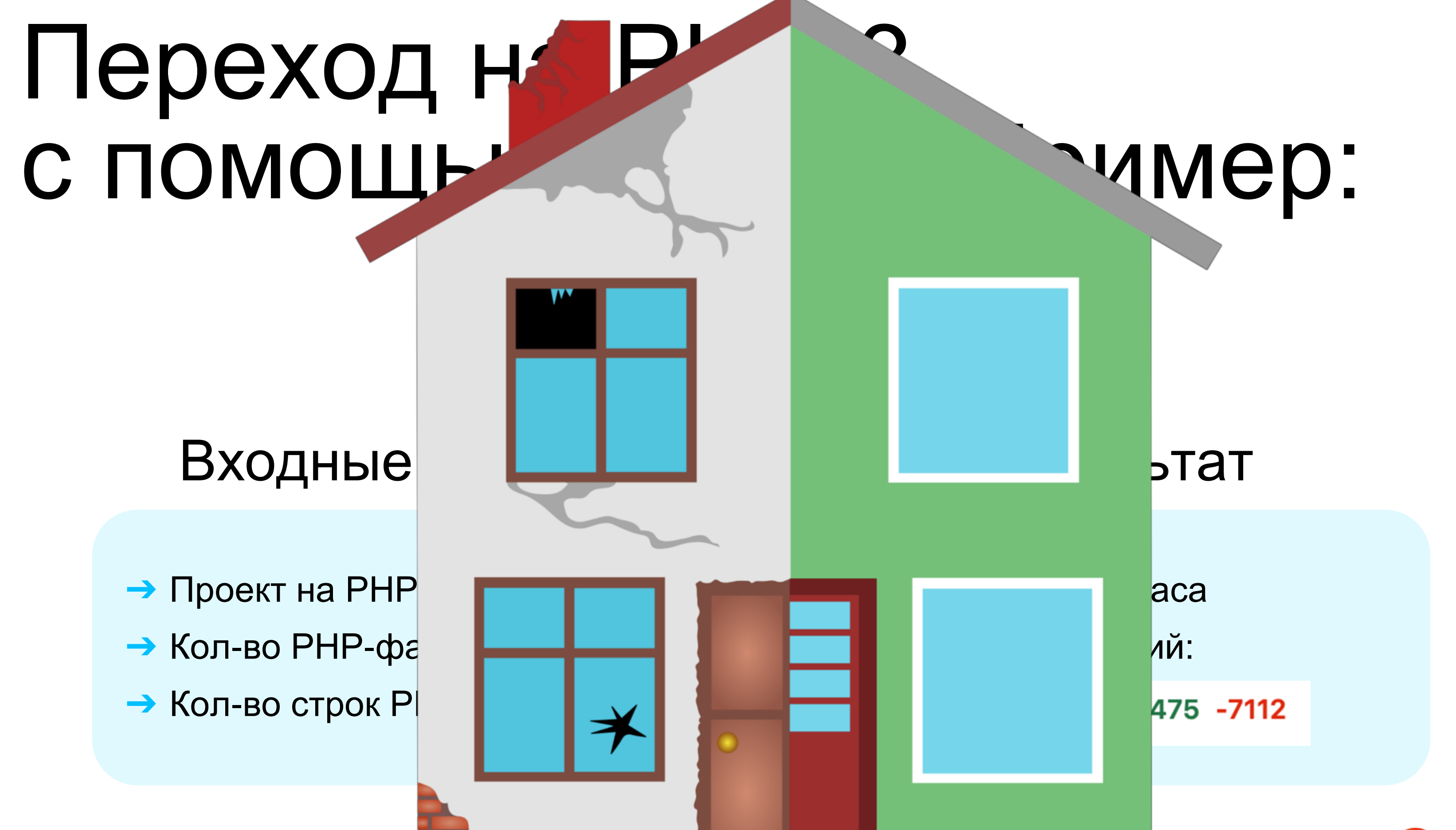


«Да как с этим работать!
Нужен рефакторинг!»

«Квартал уже запланирован.
На это нет времени»

Неутешительные выводы

- Тяжело работать с легаси-проектом.
- Нужно провести рефакторинг, но быстро.


Переход на
с помощь
имер:
Входные
ьтат
Проект на PHP
Кол-во PHP-фа
Кол-во строк PH
аса
ий:
475 -7112

Забавный факт:
На 6% сократилось количество строк кода после перехода на PHP 8

до
28 976

после
26 904



Плюсы апгрейда с помощью Rector



Простая настройка



Высокая скорость вне зависимости от количества кода

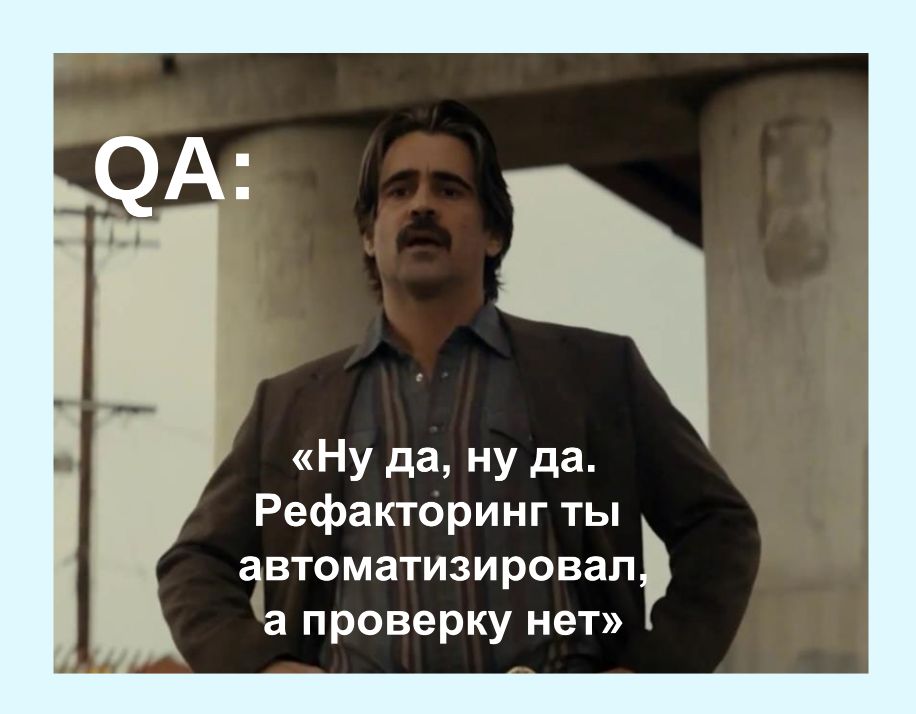


Отсутствие человеческого фактора

Нельзя так просто взять и от**rector**ить



1. Улучшаем покрытие тестами



≥50%

Всего времени рефакторинга

2. Настраиваем статический анализ



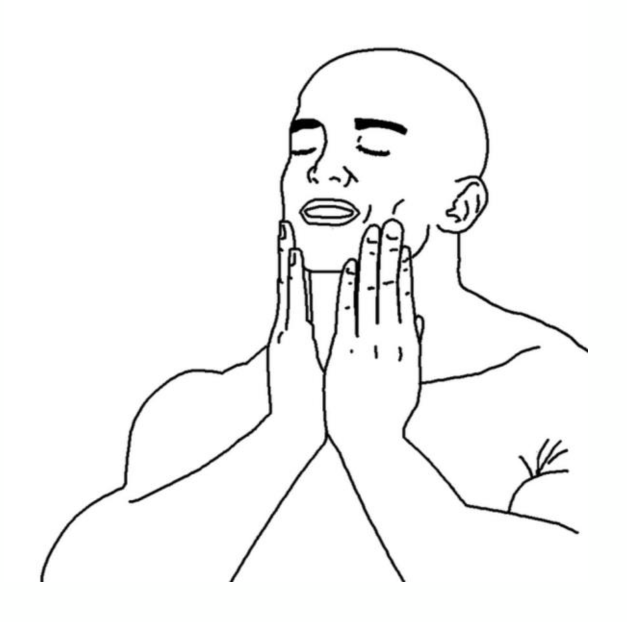
Это поможет
Rector эффективнее
проводить рефакторинг.

Достаточно **Psalm** 6-8 уровень
и/или **PHPStan** 3-4 уровень.

3. Задаем код-стайл



Rector не заботится
о простом код-стайле!



- PHP CS Fixer
- Slevomat Coding Standard
- Easy Coding Standard

Когда Rector вывел
всё в одну строку,
но код-стайл-фиксер
всё поправил



```php
<?php declare(strict_types=1);

use ...

return static function (RectorConfig $rectorConfig): void {

};
```

4. Настраиваем Rector

4.1 Указываем, что рефакторить, а что пропустить.

4.2 Включение параллельной обработки.

4.3 Настраиваем импорт имен.

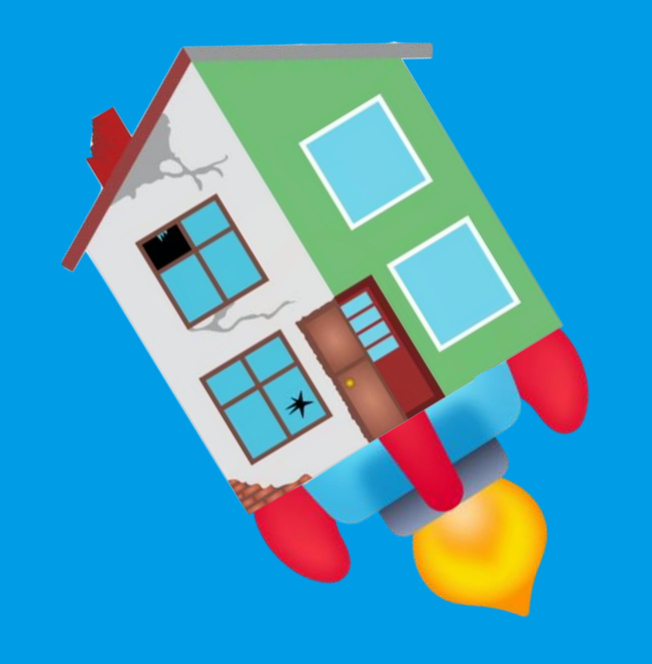
4.4 Добавляем сами правила апгрейдов.



Вывод

Предварительные работы могут отнять немало времени.

Но в конечном итоге Rector сэкономил кучу времени и сделал эту работу качественно.



Часть вторая: почему Rector?

Аналоги. Возможности.
Цели.

Аналоги

phabelio/**phabel**



PHP transpiler - Write and deploy modern PHP 8 code, today.

1 Contributor · 26 Used by · 219 Stars · 6 Forks



slevomat/**coding-standard**



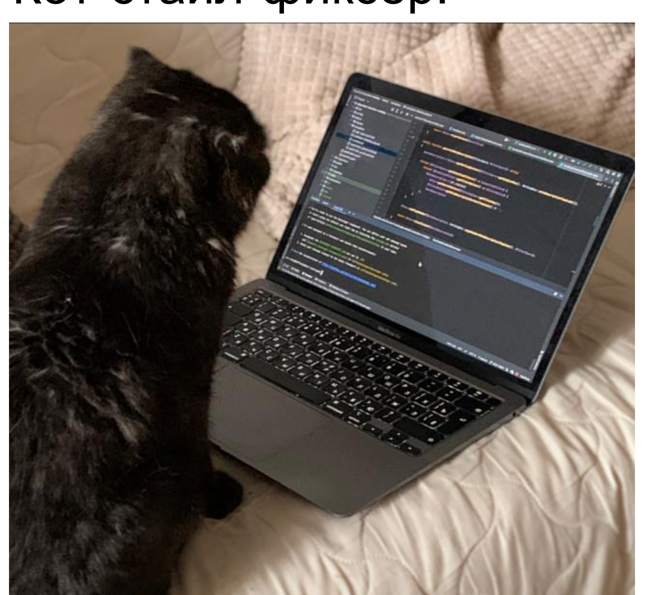
Slevomat Coding Standard for PHP_CodeSniffer provides many useful sniffs

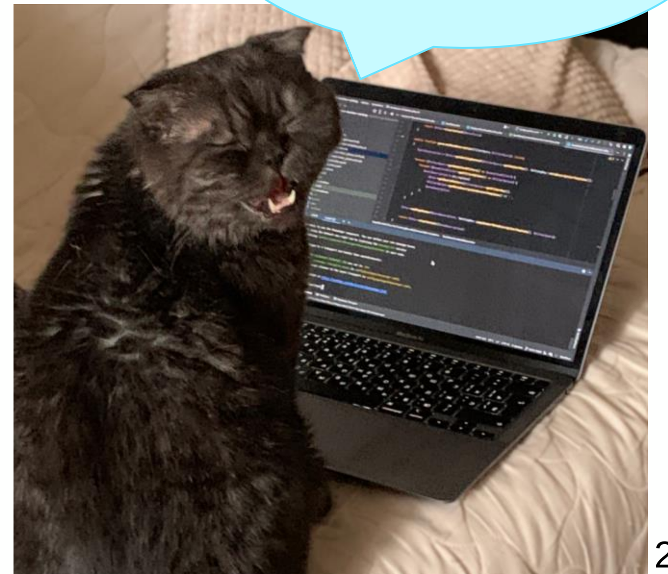
67 Contributors · 47 Issues · 1k Stars · 152 Forks



Код-стайл-фиксеры не умеют переводить на новые фичи PHP!

Кот-стайл-фиксер:



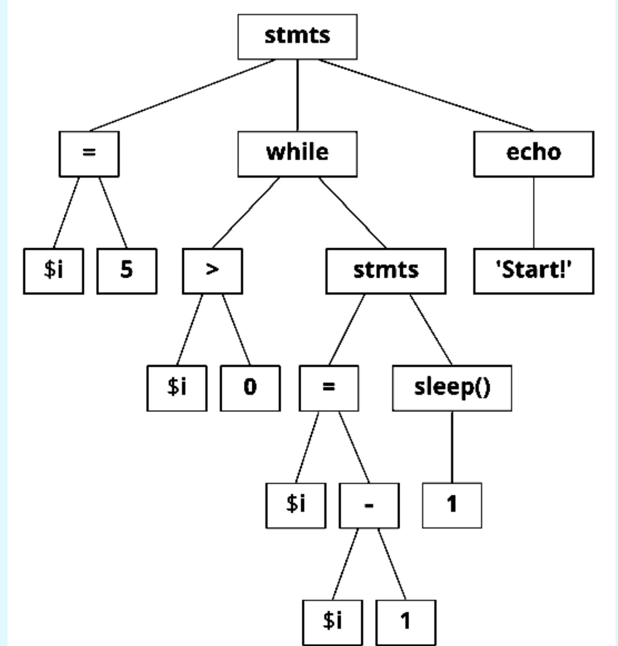


Какой ещё PHP 8?!
У меня лапки!

Код-стайл-фиксеры работают с токенами

Line	Token	Value
1	T_OPEN_TAG	<?php\n
2	T_WHITESPACE	\n
3	T_VARIABLE	$answer
3	T_WHITESPACE	
3	=	=
3	T_WHITESPACE	
3	T_CONSTANT_ENCAPSED_STRING	'Hello World!'
3	;	;
5	T_WHITESPACE	\n\n
5	T_ECHO	Echo
5	T_WHITESPACE	
5	T_VARIABLE	$answer
5	;	;

Rector работает с AST





Место Rector среди других инструментов

	Reports issues	**Fixes issues**
Coding standard	PHP CodeSniffer	PHP CS Fixer, ECS, SCS, Prettier
Logic	PHPStan, Psalm, Phan	Rector

Место Rector среди других инструментов

	Reports issues	Fixes issues
Coding standard	PHP CodeSniffer	PHP CS Fixer, ECS, SCS, Prettier
Logic	PHPStan, Psalm, Phan	Rector

Сравнение по GitHub

Rector

- Дата создания: 15.06.2017
- 6.1к ☆
- 13м установок
- Активность ежедневная

Phabel

- Дата создания: 03.08.2020
- 219 ☆
- 57к установок
- Активность слабая

Slevomat Coding Standart

- Дата создания: 18.12.2015
- 1.2к ☆
- 33м установок
- Активность нормальная

Сравнение по возможности апгрейда до PHP 8

	Атрибуты	Объявление свойств в конструк-торе	Тип Union	Тип Mixed	Выражение Match	Catch без указания переменной	Добавление ::class для объектов	Оператор Nullsafe	Конверта-ция на новые функции str_	Stringable	Соответст-вие метода сигнатуре родителя
Phabel	-	+	+	+	+	+	-	+	-	-	-
Slevomat Coding Standart	-	+	+	+	-	+	+	+	-	-	-
Rector	+	+	+	+	+	+	+	-	+	+	+

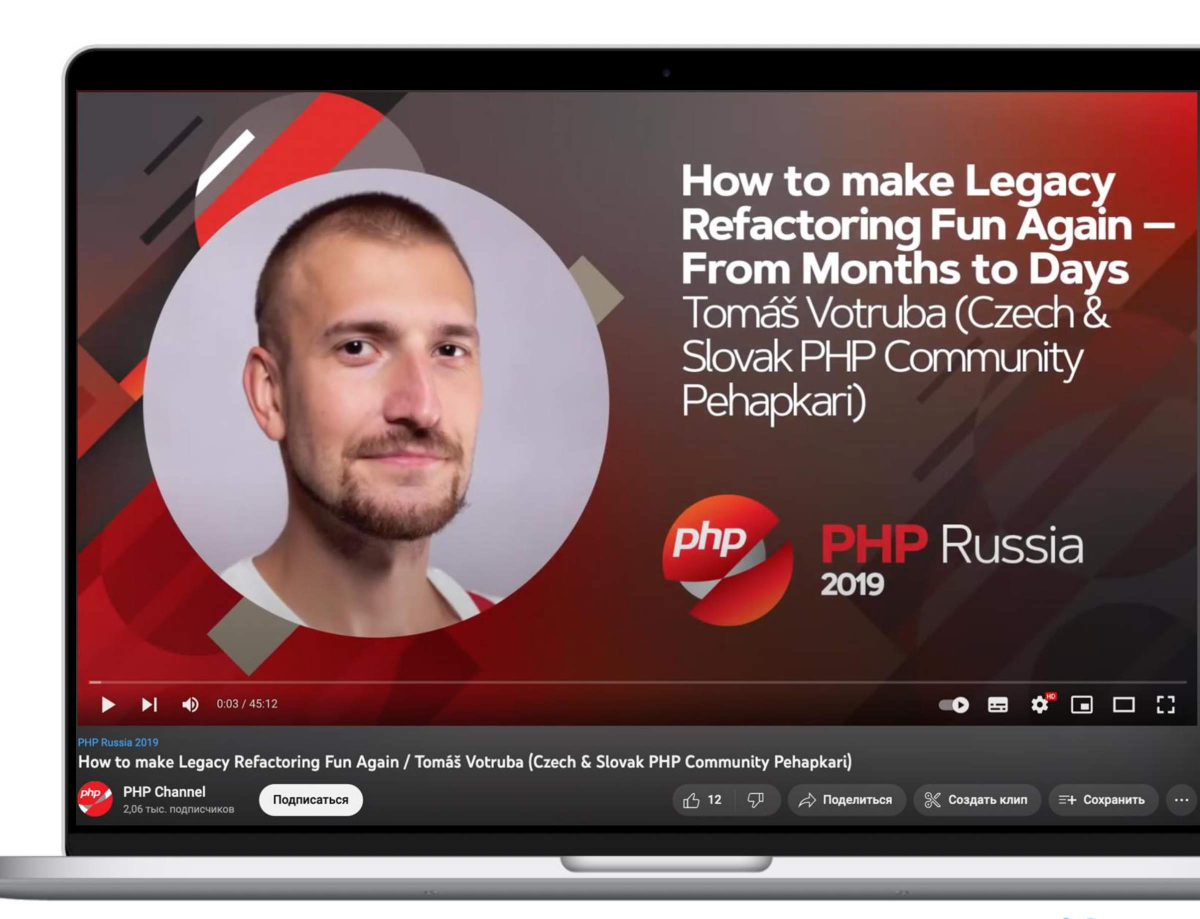
Сравнение по возможности апгрейда до PHP 8

	Атрибуты	Объявление свойств в конструк-торе	Тип Union	Тип Mixed	Выражение Match	Catch без указания переменной	Добавление ::class для объектов	Оператор Nullsafe	Конверта-ция на новые функции str_	Stringable	Соответст-вие метода сигнатуре родителя
Phabel	-	+	+	+	+	+	-	+	-	-	-
Slevomat Coding Standart	-	+	+	+	-	+	+	+	-	-	-
Rector	+	+	+	+	+	+	+	-	+	+	+

Сравнение по возможности апгрейда до PHP 8

	Атрибуты	Объявление свойств в конструк-торе	Тип Union	Тип Mixed	Выражение Match	Catch без указания переменной	Добавление ::class для объектов	Оператор Nullsafe	Конверта-ция на новые функции str_	Stringable	Соответст-вие метода сигнатуре родителя
Phabel	-	+	+	+	+	+	-	+	-	-	-
Slevomat Coding Standart	-	+	+	+	-	+	+	+	-	-	-
Rector	+	+	+	+	+	+	+	-	+	+	+

- Разработал Tomáš Votruba
- Документация, книга и блог
- Использует PhpParser и PHPStan
- Имеет 400+ правил



https://youtu.be/O35cIK3JUQA
доклад на PHP Russia 2019

Правило (Rule) — единица рефакторинга, которая изменяет конкретную часть кода.

MixedTypeRector

Change mixed docs type to mixed typed

- class: `Rector\Php80\Rector\FunctionLike\MixedTypeRector`

```
 class SomeClass
 {
-    /**
-     * @param mixed $param
-     */
-    public function run($param)
+    public function run(mixed $param)
     {
     }
 }
```

Схожие по смыслу правила объединяют в **наборы правил (Set Rules)**.

```
return static function (RectorConfig $rectorConfig): void {

    $rectorConfig->sets([
        SetList::TYPE_DECLARATION,
    ]);
};
```

Три категории правил

1. Апгрейд/даунгрейд
2. Качество кода
3. Настраиваемые

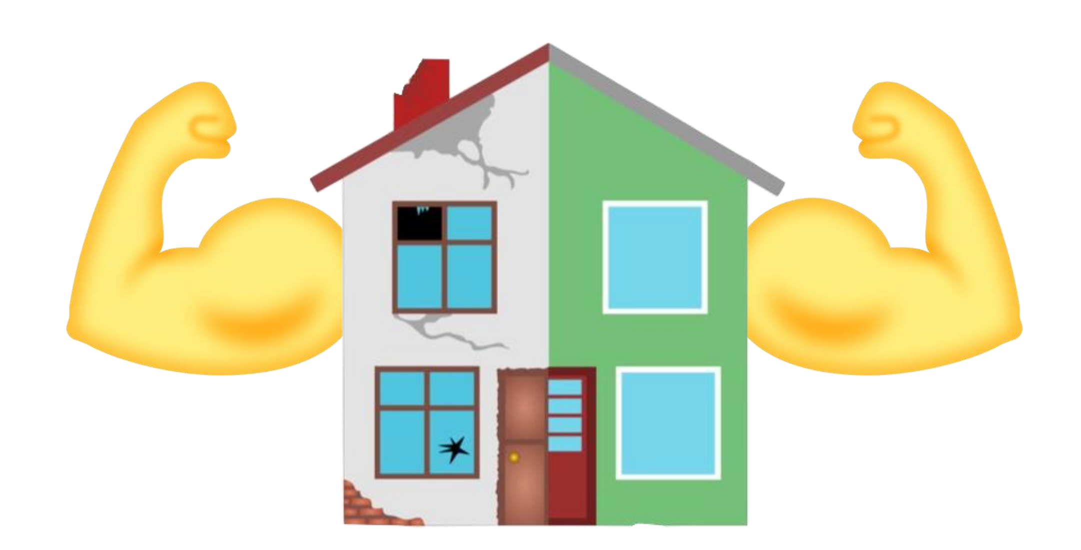
```php
$rectorConfig->sets([
    SetList::CODING_STYLE,
    SetList::CODE_QUALITY,
    SetList::TYPE_DECLARATION,
    SetList::TYPE_DECLARATION_STRICT,
    SetList::DEAD_CODE,
```

```php
$rectorConfig->ruleWithConfiguration(RenameClassConstFetchRector::class, [
    new RenameClassConstFetch(
        'App\Infrastructure\Doctrine\Paginator',
        'PAGE_SIZE',
        'ITEMS_PER_PAGE',
    ),
]);
```

```php
    SymfonySetList::SYMFONY_CONSTRUCTOR_INJECTION,
    SymfonySetList::ANNOTATIONS_TO_ATTRIBUTES,


    PHPUnitSetList::PHPUNIT_CODE_QUALITY,
    DoctrineSetList::DOCTRINE_CODE_QUALITY,
]);
```

Rector — лучший инструмент для апгрейда кода



Главная задача Rector — логический рефакторинг



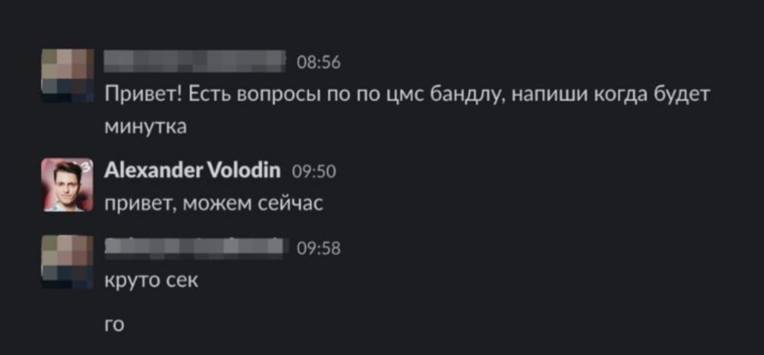
Часть третья: Rector на максималках

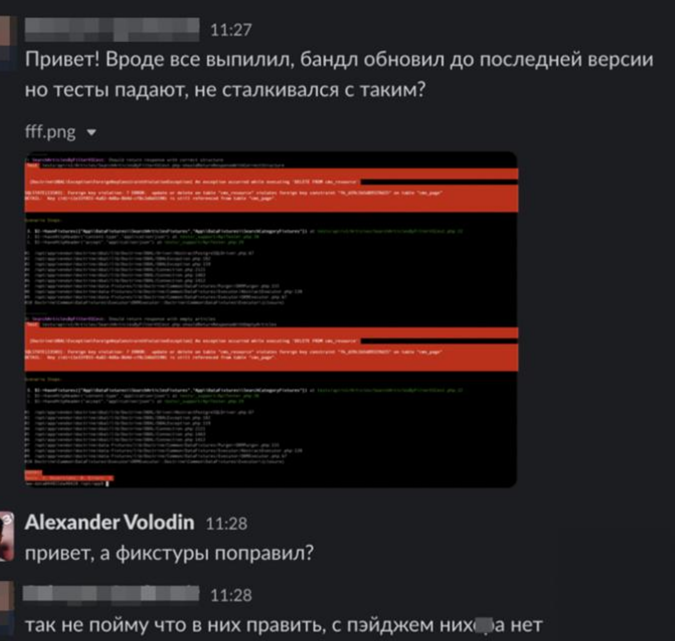
- Апгрейдить код легко! НО.
- Но как быть с апгрейдом пакетов?

История одного Бандла



Очевидное нам неочевидно другим





Проблемы обновления пакетов

Со стороны пользователя

- Изучить особенности новой версии (changelog)
- Поправить deprecated, внести новые требования версии
- Отладить работу

Со стороны мейнтейнера

- Поддержка обратной совместимости
- Растягивание выхода мажорных релизов
- Тратить время на консультации или помощь по переходу на новые версии

Основные этапы обновления Symfony-проекта

Symfony Flex

Обновление проекта с помощью рецептов symfony/flex

Rector

Обновление кода проекта с помощью наборов правил Rector'a

```
$rectorConfig->sets([
    SymfonyLevelSetList::UP_TO_SYMFONY_54,
]);
```

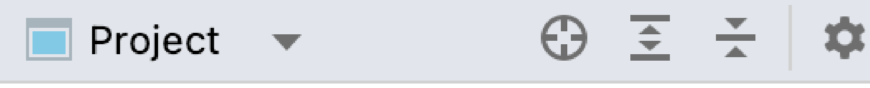


Composer

Обновление самих пакетов Symfony

```
composer update symfony/*
```



- **marketing-cms-bundle** /opt/skyeng/
  - src
  - utils
    - rector
      - config
        - sets
          - up_to_last_ver.php
          - ver_32.php
          - ver_33.php
          - ver_34.php
          - ver_40.php
      - src
        - Set
          - CmsSetList.php

1. Задаем структуру для наборов правил в нашем пакете

2. Создаем константу для набора правил

src/Set/CmsSetList.php

```php
<?php declare(strict_types=1);

namespace Skyeng\CmsBundle\Utils\Rector\Set;

use Rector\Set\Contract\SetListInterface;

class CmsSetList implements SetListInterface
{
    public const VER_32 = __DIR__.'/../../config/sets/ver_32.php';
    public const VER_33 = __DIR__.'/../../config/sets/ver_33.php';
    public const VER_34 = __DIR__.'/../../config/sets/ver_34.php';
    public const VER_40 = __DIR__.'/../../config/sets/ver_40.php';

    public const UP_TO_LAST_VER = __DIR__.'/../../config/sets/up_to_last.php';
}
```

3. Пишем набор правил config/sets/ver_40.php

```php
<?php declare(strict_types=1);

use ...

return static function (RectorConfig $rectorConfig): void {
    $rectorConfig->ruleWithConfiguration(RenameNamespaceRector::class, [
        'Skyeng\CmsBundle\UI\EasyAdminField' => 'Skyeng\CmsBundle\Infrastructure\EasyAdmin\Field',
    ]);

    $rectorConfig->ruleWithConfiguration(RenameMethodRector::class, [
        new MethodCallRename(FieldRepository::class, 'findByValue', 'findByName'),
        new MethodCallRename(FieldRepositoryInterface::class, 'findByValue', 'findByName'),
    ]);

    $rectorConfig->ruleWithConfiguration(RemoveInterfacesRector::class, [
        'Skyeng\CmsBundle\UI\ResponseInterface',
    ]);
};
```

Возможности встроенных настраиваемых правил

Переименование

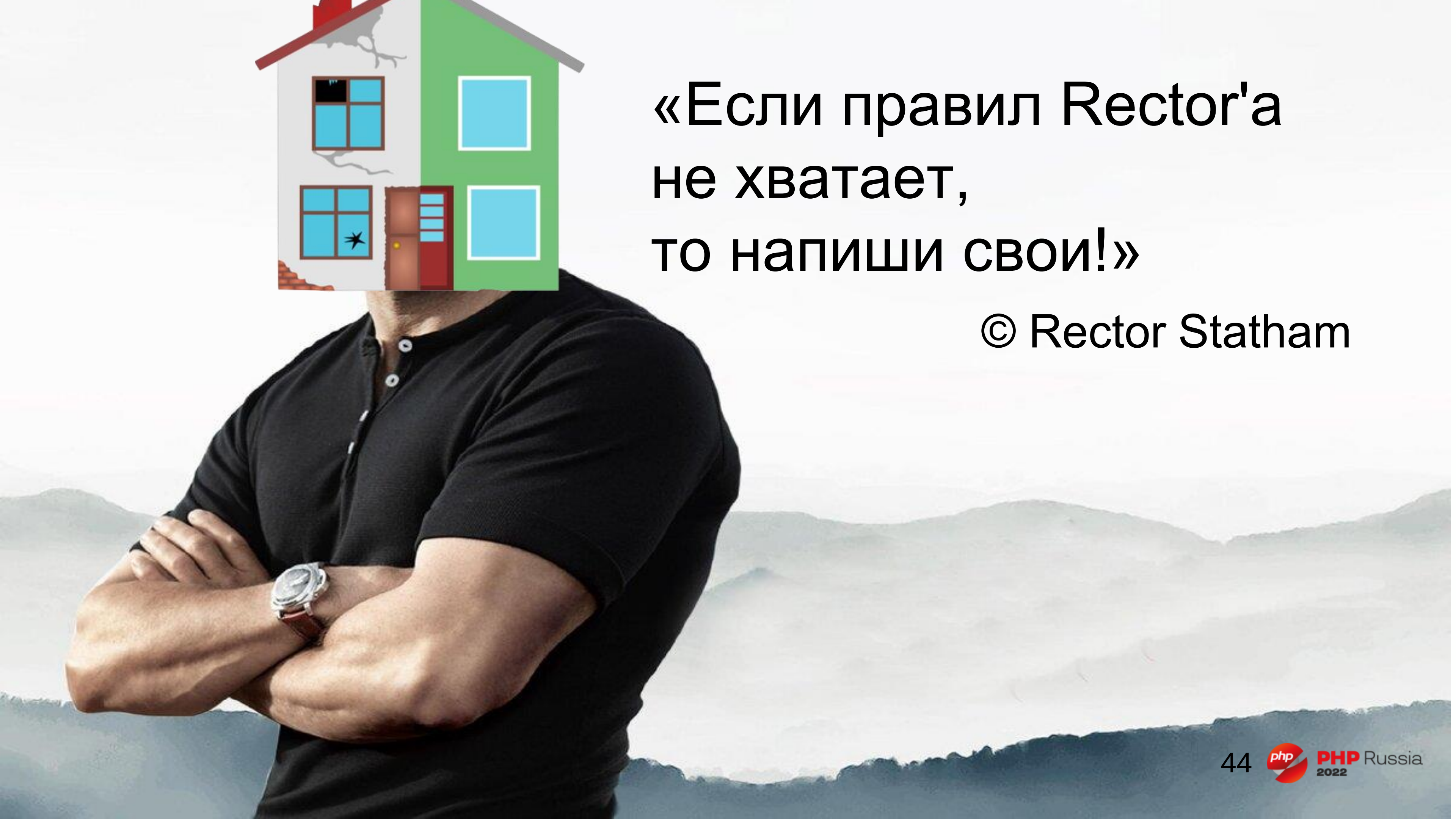
- неймспейсов
- классов
- интерфейсов
- констант
- свойств
- функций и методов

Удаление

- интерфейсов
- классов
- трейтов
- аргументов функций и методов

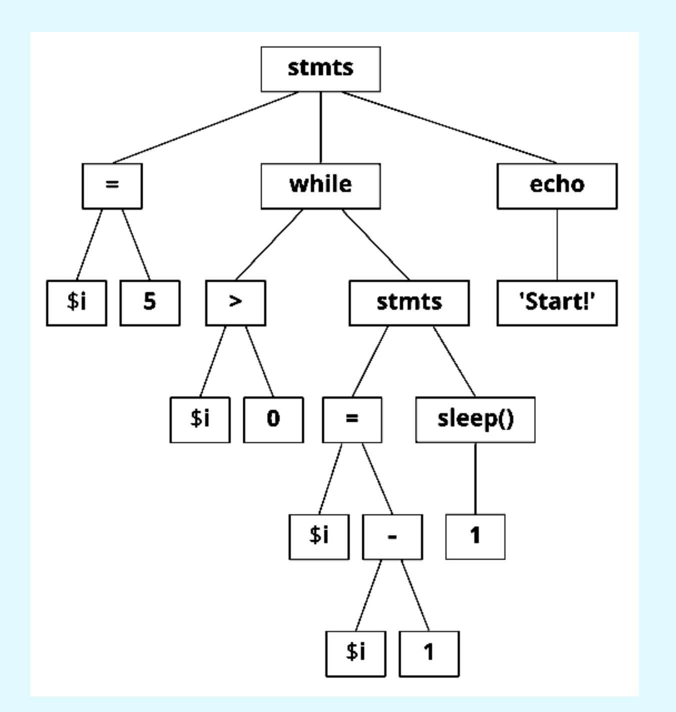
Трансформация

- замена вызова одних функций или методов на другие
- замена одних сервисов на другие
- изменение аргументов


«Если правил Rector'a
не хватает,
то напиши свои!»
© Rector Statham

Как работает Rector

- Парсинг кода в AST
- Проходится по каждому узлу
- Если тип узла подходит правилу, то обрабатывает его
- Результат обработки либо `null` (нет изменений), либо `Node` (есть изменения).



Интерфейс правил

```php
<?php declare (strict_types=1);

namespace Rector\Core\Contract\Rector;

use PhpParser\Node;

interface PhpRectorInterface
{
    /**
     * @return array<class-string<Node>>
     */
    public function getNodeTypes() : array;

    /**
     * @return Node|Node[]|null
     */
    public function refactor(Node $node);
}
```

- → Нужно реализовать `PhpRectorInterface`
- → Но лучше наследоваться от `AbstractRector`



Алгоритм работы правил

➔ Пропустить узел, если он нам не подходит

➔ Обновить/заменить текущий узел

```
class ReplaceStringWithConstRector extends AbstractRector
{
    public function getNodeTypes(): array
    {
        return [Node\Scalar\String_::class];
    }

    public function refactor(Node $node): ?Node
    {

    }
}
```

Где хранить свои правила

Project
marketing-cms-bundle /opt/skyeng/data/marketing-cms-bu
src
utils
rector
config
sets
up_to_last_ver.php
ver_32.php
ver_33.php
ver_34.php
ver_40.php
src
Rule
ReplaceStringWithConstRector.php
Set
CmsSetList.php
tests
ReplaceStringWithConstRector
config
config.php
Fixture
replace-datetime-string-with-type.php.inc
skip-if-this-is-const.php.inc
skip-if-unknown-string.php.inc
ReplaceStringWithConstRectorTest.php

Правила храним в директории `src/Rule`

Можно удобно писать unit-тесты для своих правил с помощью `AbstractRectorTestCase`




Чек-лист новой версии пакета

- ✓ Внесли правки в код пакета
- ✓ Написали набор правил для перехода на новую версию: `CmsSetRule::VER_40`
- ✓ Выпустили релиз новой версии: **4.0**



Процесс перехода на новую версию пакета

- ➔ Обновляем пакет: `composer update skyeng/cms-bundle`
- ➔ Добавляем в rector.php набор правил `CmsSetRule::VER_40`
- ➔ Запускаем Rector: `vendor/bin/rector process`

Упрощаем процесс с помощью Composer Scripts



Подписываемся на события обновления пакетов.

/composer.json

```
"scripts": {
    "post-package-update": [
        "App\\Infrastructure\\Composer\\EventHandler::postPackageUpdate"
    ],
```

/src/Infrastructure/Composer/EventHandler.php

```php
<?php declare(strict_types=1);

namespace App\Infrastructure\Composer;

use Composer\Installer\PackageEvent;
use Composer\Util\ProcessExecutor;

class EventHandler
{
    public static function postPackageUpdate(PackageEvent $event): void
    {
        if ($event->getOperation()->getTargetPackage()->getName() !== 'skyeng/cms-bundle') {
            return;
        }

        if ('y' !== $event->getIO()->ask('Execute Rector script? [y,n]', 'y')) {
            return;
        }

        (new ProcessExecutor($event->getIO()))->execute(
            'vendor/bin/rector process --config=cms-bundle-rector.php'
        );
    }
}
```

1. Проверяем, что обновился именно нужный пакет.

2. Спрашиваем пользователя, хочет ли он, чтобы Rector обновил код.

3. Запускаем Rector только с набором правил обновления пакета.

/cms-bundle-rector.php

```php
<?php declare(strict_types=1);

use App\Utils\Rector\CommonRectorConfig;
use Rector\Config\RectorConfig;
use Skyeng\CmsBundle\Utils\Rector\Set\CmsSetList;

return static function (RectorConfig $rectorConfig): void {
    CommonRectorConfig::configure($rectorConfig);

    $rectorConfig->sets([
        CmsSetList::UP_TO_LAST_VER,
    ]);
};
```

Имеет только один набор правил для перехода на последнюю версию.

Одна команда для всего процесса перехода на новую версию пакета!



Преимущества подхода

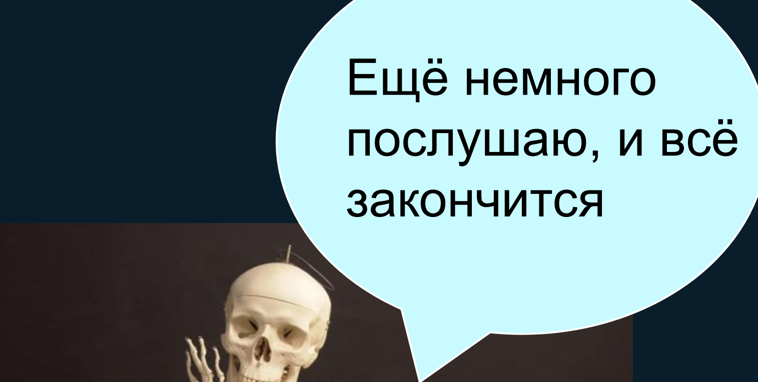
Для пользователя

- Не нужно погружаться в детали релиза
- Ускоряется переход на новую версию

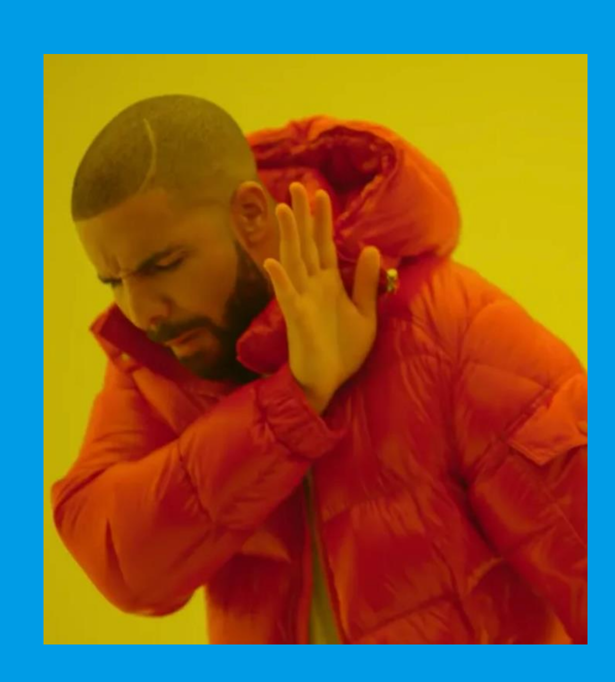
Для мейнтейнеров

- Более радикальные релизы
- Снизит необходимость обеспечивать обратную совместимость
- Меньше работы с комьюнити

Часть четвертая: Архитектурный рефакторинг

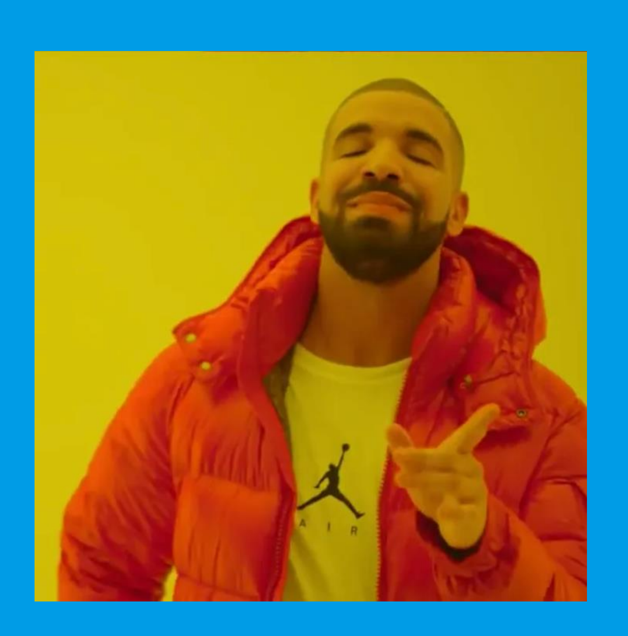


Беспорядок в проекте



```
> public
v src
  > Application
  > Command
  > Controller
  > DataFixtures
  > Domain
  > Entity
  > EventListener
  > Form
  > Infrastructure
  > Migrations
  > Repository
  > Resources
  > Service
  > SplitTest
  > UI
    Kernel.php
> tests
```

Порядок в проекте



- migrations
- public
- src
  - Application
  - Domain
  - Infrastructure
  - UI
  - Kernel.php
- tests

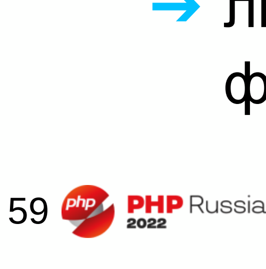


Сложно провести рефакторинг проекта, если он активно развивается.

Это приводит:

- либо к мердж-конфликтам
- либо к торможению фич-релизов

```
{
    "name": "skyeng/skyeng-frontface-backend",
    "description": "Backend for skyeng.ru",
    "type": "project",
    "license": "proprietary",
    "require": {
        "php": "^8.0",
        "ext-ctype": "*",
        "ext-dom": "*",
        "ext-gd": "*",
        "ext-iconv": "*",
        "ext-json": "*",
        "ext-simplexml": "*",
        "browscap/browscap-php": "^5.0",
        "composer/package-versions-deprecated": "^1.11",
        "damienharper/auditor-bundle": "^4.0",
        "doctrine/annotations": "^1.0",
        "doctrine/dbal": "^2.13",
        "doctrine/doctrine-bundle": "^2.3",
        "doctrine/doctrine-migrations-bundle": "^2.2",
        "doctrine/orm": "^2.8",
        "domnikl/statsd": "^3.0",
        "easycorp/easyadmin-bundle": "^3.2",
        "ekino/newrelic-bundle": "^2.2",
        "friendsofsymfony/ckeditor-bundle": "^2.2",
        "league/flysystem-aws-s3-v3": "^1.0",
        "league/flysystem-bundle": "^1.5",
        "league/uri-parser": "^1.4",
        "nelmio/api-doc-bundle": "^4.0",
        "nelmio/cors-bundle": "^2.2",
        "nikic/php-parser": "^4.8",
        "nyholm/psr7": "^1.5",
```

И здесь нам поможет Rector !



С помощью правил Rector описываем:

- ➔ что и куда хотим переместить
- ➔ что переименовать
- ➔ что удалить

Процесс похож на написание наборов правил для обновления пакета.

Создаем отдельный конфиг architect-rector.php

```php
return static function (RectorConfig $rectorConfig): void {
    $rectorConfig->paths([...]);
    $rectorConfig->importNames();
    $rectorConfig->symfonyContainerXml(__DIR__.'/var/cache/dev/App_KernelDevDebugContainer.xml');

    $rectorConfig->ruleWithConfiguration(RenameNamespaceRector::class, [
        'App\Controller\Admin'      => 'App\UI\Controller\Admin',
        'App\Command'               => 'App\UI\Command',
        'App\DataFixtures'          => 'App\Tests\DataFixtures',
        'App\Entity'                => 'App\Domain\Entity',
        'App\EventListener'         => 'App\Infrastructure\VichUploader\EventSubscriber',
        'App\Repository'            => 'App\Infrastructure\Doctrine\Repository',
        'App\Service'               => 'App\Application',
        'App\Application\Articles'  => 'App\Application\Article',
    ]);

    $rectorConfig->rule(EventListenerToEventSubscriberRector::class);

    $rectorConfig->ruleWithConfiguration(RenameClassRector::class, [
        'App\Form\CkEditorField'       => 'App\Infrastructure\EasyAdmin\Field\CkEditorField',
        'App\Form\VichImageField'      => 'App\Infrastructure\EasyAdmin\Field\VichImageField',
        'App\Form\FavoriteArticleType' => 'App\Infrastructure\Symfony\Form\Type\FavoriteArticleType',
        'App\Form\ImageType'           => 'App\Infrastructure\Symfony\Form\Type\ImageType',
    ]);
};
```

Файл с описанием команд манипуляций над файлами и директориями

```yaml
tasks:
    refactor:
        desc: Команда для рефактоиринга
        cmds:
            - vendor/bin/rector process --config=architect-rector.php --clear-cache

            - mkdir -p ./tests/_support/DataFixtures
            - mkdir -p ./src/Infrastructure/VichUploader/EventSubscriber

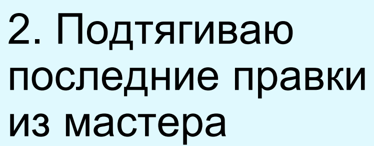
            - rsync -av ./src/Controller/Admin/*            ./src/UI/Controller/Admin
            - rsync -av ./src/Controller/Api/V1/*           ./src/UI/Controller/Api
            - rsync -av ./src/Command/*                     ./src/UI/Command
            - rsync -av ./src/DataFixtures/*                ./tests/_support/DataFixtures
            - rsync -av ./src/Entity/*                      ./src/Domain/Entity
            - rsync -av ./src/Repository/*                  ./src/Domain/Repository
            - rsync -av ./src/EventListener/*               ./src/Infrastructure/VichUploader/EventSubscriber
            - rsync -av ./src/Migrations/*                  ./migrations
            - rsync -av ./src/Service/*                     ./src/Application
            - rsync -av ./src/Application/Articles/*        ./src/Application/Article
            - rsync -av ./src/Form/CkEditorField.php        ./src/Infrastructure/EasyAdmin/Field/CkEditorField.php
            - rsync -av ./src/Form/VichImageField.php       ./src/Infrastructure/EasyAdmin/Field/VichImageField.php
            - rsync -av ./src/Form/ImageType.php            ./src/Infrastructure/Symfony/Form/Type/ImageType.php

            - find . -type d -empty -delete
```

День рефакторинга

1. Оповещаю команду, чтобы сегодня не трогали проект





2. Подтягиваю последние правки из мастера

3. Выполняю команду:
> task refactor

4. Тестирование



5. Релиз на проде

Результаты

1

Сократили время, при котором команда не может трогать проект, до 1 дня

2

Можно спокойно откатывать правки по рефакторингу

3

Качество и надежность — убираем человеческий фактор

4

И можем пере-использовать в аналогичном проекте

Подведем итоги



Когда с коллегой смотришь доклад онлайн

Итоги

- Rector отлично подходит для апгрейда кода
- Может служить как дополнительный анализатор/фиксер качества кода
- Улучшит процесс обновления пакетов

Когда используешь Rector:

Для обновления кода

Для поддержки качества кода

Для обновления пакетов





Итоги

→ Также Rector может помочь при архитектурном рефакторинге

Когда используешь Rector:

Для поддержки качества кода

Для обновления пакетов

Для архитектурного рефакторинга







Для борьбы с новыми проблемами
будут создаваться практики,

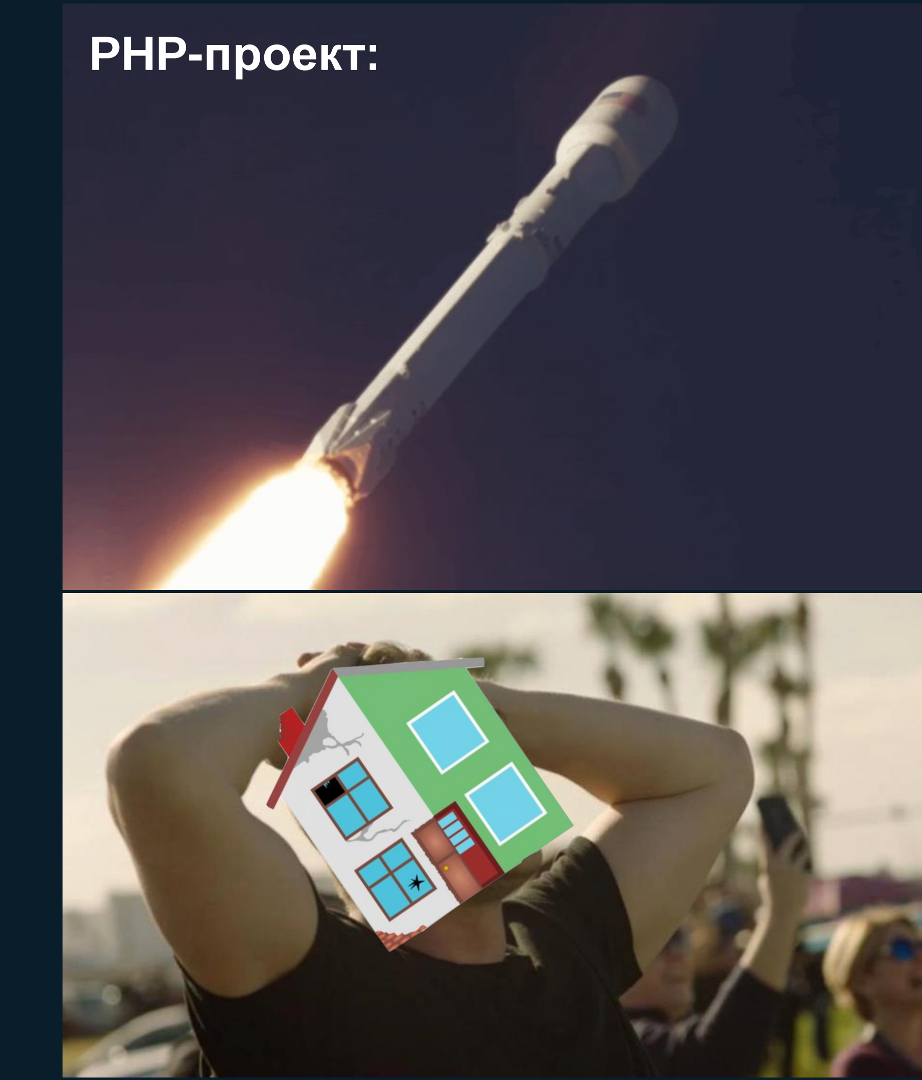
… а эти практики будут автоматизироваться
с помощью правил Rector’a.

Чем больше проблем будет покрыто автоматическим рефакторингом —

…тем проще избавиться от легаси.

И тогда мы будем больше внимания уделять фичам,

…и быстрее двигаться в будущее! :)



Оценить доклад
и дать обратную связь





PHP Russia
2022

Спасибо!

Автор доклада: **Александр Володин**

Контакты:
alex.volodin.work@gmail.com
t.me/sasha_proger

Приложение к докладу:
github.com/alex-volodin/phprussia

Ссылка на презентацию:
clck.ru/32kv6H